REMARKS and ARGUMENTS

Claims 1-18, 20-28, 32-51, and 55-65 are pending in the Application. Claims 1, 17, 35, 51, and 55 are independent claims. All claims have been rejected.

IDS

An IDS is being filed herewith, which correctly indicates the publication numbers for application by Cota-Robles and Williams et al. Applicants request that such applications be considered during Examination.

Title Objection

The Office Action has objected to the Title at being non-descriptive but has not provided Applicants with any insight into why the title is found to be non-description and has further declined to provide a suggested amendment to the Title. Applicants have provided herein a suggested title amendment: A Method and System to Provide Concurrent User-Level, Non-Privileged Shared Resource Thread Creation and Execution ~~Multithreading~~. The objection has been overcome and should be withdrawn.

Claim Objections

The Office Action has objected to Claim 44 for lack of antecedent basis for "the multiprocessor." Such language has been changed to reflect "the machine" as recited in Claim 35, from which Claim 44 depends. The objection to Claim 44 has thus been overcome, and should be withdrawn.

The Office Action has also objected to Claim 61 for lack of antecedent basis for "said prioritizer." Claim 61 has been amended to correct a dependence error. Claim 61 has been

amended to depend from Claim 34 (which recites the prioritizer) rather than Claim 33. The objection to Claim 61 has thus been overcome and should be withdrawn.

Claim Rejections – 35 USC § 112

Claim 3 stands rejected under 35 U.S.C. § 112, second paragraph, as being indefinite in its use of "a first thread", "a second thread" and "a second shred". The Applicants have clearly indicated a distinction between shreds and threads in the Application. For example, they state at paragraph 28 that: "Multiple threads of execution are distinguishable from existing shared-memory threads, and are referred to as *shreds*, or shared resource threads." In addition, in the comparison chart in Table 1 at paragraph 30 of the Application, Applicants provide additional distinction between "threads" (Shared Memory Multiprocessor Threads) and "shreds". Applicants state that a shared memory multiprocessor thread is created and destroyed by an operating system call whereas a shred is created and destroyed by a non-privileged instruction. A thread communicates with other threads via shared memory, whereas shreds may communicate with either other via shared registers as well as shared memory. Applicants also state that threads each maintain a unique system state whereas the system state for all shreds [that are associated with a particular OS-created thread] share a system state. Based at least in part on these portions of the Application, the language of Claim 3 has been amended to make the distinction between the first thread, second thread and second shred more clear. The rejection of Claim 3 has thus been overcome and the rejection should be withdrawn.

Claim Rejections -35 USC § 102(b)

The Office Action has rejected Claims 17-18, 20-28, 32, 35-44, 55-58 and 62 under 35 U.S.C. § 102(b) as being anticipated by U.S. Pat. No. 5,485,626 to Lawlor et al. (referred to

hereinafter as "Lawlor"). However, the Office Action has failed to make a prima facie case of anticipation for the claims, and such rejections should be withdrawn.

"[F]or anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention ..." MPEP 706.02 (emphasis added). "The identical invention must be shown *in as complete detail as contained in the ... claim.*" *Richardson v., Suzuki Motor Co.*, 868 F. 2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added). Lawlor simply fails to disclose every aspect of the claimed invention. The Examiner has therefore failed to meet his burden of making a prima facie case of anticipation.

Lawlor discloses an architecture that utilizes "encapsulation" in which a system controls object storage for queues and counters, such that a user cannot directly access the object storage at any time. (Lawlor, Abstract). That is, " [i]f the user needs to access a queue, the user must request if from the system." (Lawlor, Col. 7, lines 20-21). "[T]he user cannot access it directly at any time." (Lawlor, Col. 7, lines 19-20). If the user needs to access a queue, the user must request it from the system, and the user is issued a "token". Through this use of tokens, threads can access the objects to permit communications among threads. (Lawlor, Col. 7, lines 21-34). An object dispatcher controls the tokens such that the operating system dispatch process is not invoked. "[T]he "object storage" in which the counters and queues are held is not accessible via normal instructions." (Lawlor, Col. 8, lines 16-19).

Claim 17. A prima facie case of anticipation has not been made out with respect to Claim 17. To establish such a case, the cited reference must show the claimed invention in as much detail as is described in the claim. However, Lawlor does not disclose, teach or suggest "a shared register that is addressable by a user-level instruction, the shared register to provide communication between the shred and the one or more other shreds." (Amended Claim 17, in

part). The Lawlor teaching of encapsulated queues for thread communication, where the queues are in "object storage" that is **"not accessible via normal instructions" does not read on "a shared register that is addressable by a user-level instruction"**. Claim 17 is allowable for at least this reason. In addition, Claims 18, 20-28, 32-35, and 59-61, which depend from Claim 17, are also allowable for at least this reason.

Claim 35. A prima facie case of anticipation is not made out with respect to amended Claim 35, either. Claim 35 recites, in part, "a plurality of ~~shared resource~~ threads of execution~~(shreds)~~ that each shares a system state with an OS-generated thread " (amended Claim 35, in part). Lawlor does not disclose, teach or suggest such element.

The current amendments to Claim 35 are supported by the Specification. For example, the Specification indicates that each shred maintains its own private state that may include, for example, instruction pointer (see private state 610 of Fig. 6 and Table 4 in paragraph 36 of the Application).

However, each shred also shares system and application state with other shreds associated with the same OS-generated thread. This system and application state is shared among shreds but is private to each thread (that is, this state is not shared among different OS-generated threads). This concept is further discussed in the Application connection with Fig. 6 (see shared application and system state 620) and in Table 4 of paragraph 36 of the Application. This concept is also shown in Table 1 of the Specification at paragraph 30, which indicates that the system state for shreds is shared for all shreds but that, for a traditional OS-generated thread, a system state is maintained for each thread and is unique to that thread. Thus, the type of system shown in Lawlor is distinguished by the Applicants.

This idea of application and system state that is shared among multiple shreds but is private to each thread (see Table 4 of Application at paragraph 36) is not shown by Lawlor. Instead, Lawlor shows unique application state for each thread (see discussion of establishing "parallel application environment" at Col. 8, lines 33-46 of Lawlor). Thus, Lawlor does not suggest, teach or disclose "a plurality of ~~shared resource~~ threads of execution~~(shreds)~~ that each shares a system state with an OS-generated thread " (amended Claim 35, in part).

Accordingly, a prima facie case of anticipation has not been made out with respect to Claim 35. Claim 35 is allowable for at least these reasons. Claims 36-50 and 62, which depend from Claim 35, are also allowable for at least the foregoing reasons.

Claim 55. A prima facie case of anticipation is not made out with respect to amended Claim 55, either. Claim 55 recites, in part, "wherein the registers are addressable by one or more user-level instructions in each of a plurality of user-level threads" (amended Claim 55, in part). Lawlor does not disclose, teach or suggest such element. Instead, Lawlor explicitly teaches that queues and counters used to allow communication among threads cannot be directly accessed by the user at any time. (See Lawlor, Col. 7, lines 19-20). This is made clear at Col. 19, lines 15-34 of Lawlor, which indicates that the encapsulated SRM messages that allow "register-to-register communication between threads" (lines 31-32) are in the encapsulated object storage. Thus, the communication queues of Lawlor are not "addressable by one or more user-level instructions" (amended Claim 55, in part). Accordingly, a prima facie case of anticipation has not been made out with respect to Claim 55. Claim 55 is allowable for at least these reasons. Claims 56-58, which depend from Claim 55, are also allowable for at least the foregoing reasons.

Claim Rejections -35 USC § 103(a)

The Office Action has rejected Claims 1-12, 15-16 and 45-46 under 35 U.S.C. § 103(a) as being unpatentable over Lawlor in view of an on-line dictionary definition of the word "virtual memory" at the web site of the Free On-Line Dictionary of Computing ("Foldoc").

In addition, the Office Action has rejected 51 and 63-65 under 35 U.S.C. § 103(a) as being unpatentable over Lawlor in view of a textbook on Computer Architecture ("Computer Architecture: A Quantitative Approach") by Patterson et al. (hereinafter "Patterson"). However, the Office Action has failed to make a prima facie case of obviousness for the claims, and such rejections should be withdrawn.

The legal requirements for a prima facie case of obviousness are clear. "The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness." MPEP § 2142. It is well established that *prima facie* obviousness is only established when three basic criteria are met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (MPEP 2144). The Office Action has failed to meet one or more of these requirements.

Claim 1. Claim 1 recites, in part, "a first shared resource thread (shred) that ~~shares virtual memory address space with~~ belongs to the same process as one or more other shreds" (Amended Claim 1, in part). Neither Lawlor nor Foldoc, either alone or in combination, discloses or teaches or suggests such limitation. Foldoc provides a basic definition of the word "virtual memory". Foldoc does not teach, suggest or disclose "a first shared resource thread

(shred) that ~~shares virtual memory address space with~~ belongs to the same process as one or more other shreds." (Amended Claim 1, in part). The only reference to multitasking in Foldoc is a statement the demand paging or an MMU may be used to avoid allocation of large amounts of physical memory for each active program and run-time allocation, respectively. Foldoc also mentions that a multitasking system may use a timeslice memory swapping scheme instead of paging. Both of these references to multitasking in the Foldoc definition assume that each task or process is a distinct thread. There is simply no teaching toward multiple shreds that are associated with the same task or process.

Accordingly, a prima facie rejection of Claim 1 is not made out. Claim 1 is therefore allowable for at least this reason. In addition, Claims 15-16, which depend from Claim 1, are also allowable for at least this reason.

Claim 51. Claim 51 has not been amended because the Office Action has failed to make out a prima facie case of obviousness regarding the current claim language. Claim 51 recites, in part: "the microprocessor capable of executing multiple concurrent shreds" (Claim 51, in part). The Office Action attempts to combine together two references in order to make out a prima facie case of obviousness with respect to this claim. However, this attempt is unavailing for several reasons.

First, none of the references teaches a microprocessor capable of executing multiple concurrent threads, wherein the ISA of the microprocessor includes user-level multithreading instructions. The Office Action admits that Lawlor doesn't show a microprocessor capable of executing multiple concurrent shreds. Patterson does not disclose the element either. Patterson discloses that parallelism in a single program may be exploited by a multiple-issue processor that

allows multiple instructions to issue per clock cycle. While Patterson does teach that parallelism within a single program can be identified and dispatched in a scalar fashion, it simply does not disclose, suggest nor teach the execution of multiple *shreds* concurrently. **Scalar execution of multiple instructions per cycle does not teach concurrent execution of multiple shared resource threads ("shreds").** Because neither reference shows this element, their combination also fails to disclose this element. A prima facie case has not been made out with respect to Claim 51.

Neither of the references individually shows "the microprocessor capable of executing multiple concurrent shreds." Accordingly, combination of the references doesn't show it either. Claim 51 is allowable for at least this reason.

Even if Patterson did show this element (which Applicants strenuously deny), a prima facie case has not been made out for a second reason. That is, there is no proper showing of motivation to combine the Lawlor and Patterson references. The Office Action states that the Lawlor description of its FTDE feature shows a motivation to combine the references. This is simply wrong. The FTDE feature of Lawlor describes dynamic recruitment of available processors in a CMP environment "with many processors." This manner of improving performance is simply is not applicable to the single-processor system claimed in Claim 51. There is simply not a sufficient motivation rationale stated in the Office Action. The Examiner is improperly using hindsight.

Thus, none of the references includes a motivation to combine it with the teachings of the other references. Thus, the strained attempt to make out a prima facie case of obviousness must fail. The Examiner is asserting a rationale to combine that relies on the benefit of hindsight reasoning, rather than deduction from the prior art. Here, the asserted rationale to combine is

nowhere supported within the prior art, nor is any specific thing within the prior art cited to support the rationale. Under these circumstances, a prima facie rejection of Claim 51 is not made out. Claim 51 is therefore allowable for at least this reason. In addition, Claims 63-65, which depend from Claim 51, are also allowable for at least this reason.

Accordingly, Applicants respectfully submit that the applicable rejections have been overcome and must all be withdrawn. Applicants reserve all rights with respect to the application of the doctrine equivalents. Applicant respectfully requests that a timely Notice of Allowance be issued in this case. If the Examiner feels that an interview would help to resolve any remaining issues in the case, the Examiner is invited to contact Shireen Bacon of Intel, at (512) 732-3917.

Please charge any shortages and credit any overcharges to our Deposit Account No. 02-2666.

Respectfully submitted,

Dated: September 8, 2006

/Shireen Irani Bacon/

Shireen Irani Bacon
Reg. No. 40,494
Tel.:(512) 732-3917

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026